CLAIMS

What is claimed is:

1.   A system that facilitates specifying and utilizing hardware functionality, comprising:

a common hardware register pseudo-language, the language comprising a set of primitives; and

a specification component that specifies hardware functionality *via* the common hardware register pseudo-language.

2.   The system of claim1, the set of primitives that are loaded prior to at least one of: boot-up and during initialization.

3.   The system of claim 2, the set of primitives is concurrently loaded with an advanced configuration and power interface (ACPI) table.

4.   The system of claim 1, further comprising a linking component that links instructions to effect a particular higher-level functionality.

5.   The system of claim 4, at least one of the instructions is associated with a plurality of registers.

6.   The system of claim 4, the instructions comprise primitive(s) and corresponding resources with respect to a particular action.

7.   The system of claim 4, the instructions is unique dependent upon a vendor's proprietary hardware.

8.   The system of claim 1, a subset of the primitives effects reading from a hardware register.

29

9. The system of claim 1, a subset of the primitives effects writing to a hardware register.

10. The system of claim 1, the set of primitives includes at least bit-masked reading from a hardware register.

11. The system of claim 1, the set of primitives includes at least bit-masked writing to a hardware register.

12. The system of claim 1, the pseudo-language is a complete set of possible operations that can be performed upon a generic hardware register.

13. The system of claim 1, further comprising a common driver that supports functionality of the hardware.

14. The system of claim 1, further comprising an artificial intelligence (AI) component that infers characteristics of a hardware device.

15. The system of claim 14, the AI component infers a series of instructions to perform a high level action.

16. The system of claim 15, the instructions comprise a primitive and resources to perform a primitive.

17. The system of claim 15, the AI component comprises an implicitly trained classifier.

18. The system of claim 15, the AI component performs a probabilistic-based utility analysis in connection with loading the set of primitives.

19.   A method that specifies hardware functionality, comprising:

determining a hardware device type; and

loading a series of instructions prior to operating system kernel availability

wherein the instructions comprise at least primitives that are defined by a common

hardware register pseudo-language.

20.   The method of claim 19, loading an advanced configuration and power interface

(ACPI) table concurrent with the instructions.

21.   The method of claim 19, linking the series of instructions to effectuate a high

level action.

22.   The method of claim 21, performing the series of instructions upon at least one

hardware register.

23.   The method of claim 21, performing at least one instruction with more than one

register.

24.   The method of claim 19, inferring the series of instructions to perform a high

level action with artificial intelligence.

25.   A system specifying hardware functionality, comprising:

means for specifying a hardware functionality; and

means for linking instructions that comprise at least hardware register primitives,

wherein the primitives are defined by a common hardware register pseudo-language.

26.   The system of claim 25, further comprising means for concurrently loading the

instructions with an advanced configuration and power interface (ACPI) table.

27.   The system of claim 25, further comprising means for linking a subset of the

instructions to effect a particular higher-level functionality.

28. The system of claim 25, further comprising means for associating at least one instruction with a plurality of registers.

29. The system of claim 25 further comprising means for describing the instruction as primitive operation(s) and corresponding resources with respect to a particular action.

30. The system of claim 25, further comprising means for inferring characteristics of a hardware device.

31. The system of claim 30, further comprising means for assembling the set of primitives.

32. The system of claim 30 further comprising means for performing a probabilistic-based utility analysis in connection with executing the set of instructions.

33. A computer readable medium having stored thereon the computer-executable component(s) of claim 1.

34. A computer readable medium having stored thereon the computer-executable instructions for implementing the method of claim 19.

35. A computer readable medium having stored thereon the system of claim 25.

36. A data packet that can pass between a plurality of computer processes, comprising:

a data field storing a component that specifies hardware functionality *via* a common hardware register pseudo-language, the language comprising:

a set of primitives that are loaded prior to at least one of: boot-up and during initialization.